APPLICATION FOR UNITED STATES LETTERS PATENT

INVENTORS: Myung Sub SIM

TITLE: DEVICE AND METHOD FOR DECODING TURBO CODES

ATTORNEYS & ADDRESS: FLASHER & KIM, LLP
P. O. Box 221200
Chantilly, VA 20153-1200

DOCKET NO.: K-0329

DEVICE AND METHOD FOR DECODING TURBO CODES

BACKGROUND OF THE INVENTION

1. Field of the Invention

[1] The present invention relates to a mobile communication system and, more particularly, to a method of decoding turbo codes using a sliding window method.

2. Background of the Related Art

[2] As is well known, turbo codes are generated by two recursive systematic convolutional encoders (RSCs) connected in parallel through an internal interleaver, and this coding method is used for transmitting data of a high data rate in the next- generation mobile communication standard (3GPP or 3GPP2).

[3] The turbo code processes a generated information bit sequence in the unit of a block. Especially in case of encoding a large information bit sequence, it is known that a very superior coding gain is obtained with respect to the convolutional codes and a very superior error correction capability is achieved by iteratively decoding simple component codes in a receiving end.

[4] Recently, there has been proposed a relatively simple turbo decoding technique capable of supporting a high-speed data transmission under the mobile communication environment. In this structure, input code words alternately pass through two convolutional decoders and the complexity of the structure is greatly reduced.

[5] However, in order to iteratively pass through the convolutional decoder, it is required that the outputs of the convolutional decoder are not hard-decision values of "0" or "1", but are soft-decision values corresponding to the rate of a probability that the output of the convolutional decoder is "0" or "1."

[6] For this, there has been proposed a Maximum A Posteriori (MAP) decoding technique that calculates *a posteriori* probability values of information bits and performs the decoding so that the probability values become maximum.

[7] Generally, an information source of the turbo codes is the "Markov process" that has a discontinuous time and quantified state. Accordingly, the information source can be expressed through a binary matrix diagram.

[8] In the binary matrix, S_k represents the state of an encoder at a time k and $x_k=x_{k,1},k,x_{k,n}$ ($x_k=\{0,1\}$) represents an output of the encoder whose code rate is 1/n. Here, the number of states $S_k=m$ (m=0,1,2,..,M-1) of the information source is M.

[9] When the time is shifted from k-1 to k, an input bit d_k of the turbo encoder changes the state S_k-1 of the encoder to S_k. A state sequence $S=(S_0,..,S_T)$ of the information starts at time k=0 and ends at time k=T. The initial state S_0 of the encoder is 0.

[10] The output sequence x of the turbo encoder is modulated to BPSK or QPSK and suffers fading in a discrete memory channel. Accordingly, the sequence received in the receiving end becomes $y=(y_1,k,y_k,k,y_T)$. Here, $y_k=(y_{k,1},k,y_{k,n})$.

[11] As described above, the MAP algorithm is an algorithm for estimating the *a posteriori* probability of state shift of the information using the received sequence. The MAP

algorithm calculates the *a posteriori* probability of information bits $P(d_k = 1 \mid y)$ and $P(d_k = 0 \mid y)$. Then, the output of the decoder can be finally obtained in the form of a desired log likelihood ratio (LLR), as expressed by equation 1.

[12] [Equation 1]

$$L(d_k) = \log \frac{P(d_k = 1 \mid y)}{P(d_k = 0 \mid y)}$$

$$= \log \frac{\sum_{(m',m)\, d_k = 1} P(S_{k-1} = m', S_k = m, y)}{\sum_{(m',m)\, d_k = 0} P(S_{k-1} = m', S_k = m, y)}$$

The *a posteriori* probability $P(S_{k-1} = m', S_k = m, y)$ of the state shift of the information bits is obtained by the equation 2.

[13] [Equation 2]

$$P(S_{k-1} = m', S_k = m, y) = P(S_{k-1} = m', y_{j<k})\, P(y_{j>k} \mid S_k = m)\, P(S_k = m', y_k \mid S_{k-1} = m')$$

[14] In equation 2, $y_{j<k}$ represents the received sequence from the initial time to time k-1 and $y_{j>k}$ represents the received sequence from time k+1 to the last time.

[15] Also in equation 2, $P(S_{k-1} = m', y_{j<k})$ is defined as $\alpha(S_k\text{-}1)$ and $\alpha(S_k\text{-}1)$ is defined therefrom. $P(y_{j>k} \mid S_k = m)$ is defined as $\beta(S_k)$.

[16] In order to obtain the optimum *a posteriori* probability, a predetermined period is required before $\beta(S_k)$ is obtained. This is called a learning period. $\beta(S_k)$ calculated after the learning period is used for determination of the output bits of the decoder.

[17] Hereinafter, $\alpha(S_k)$ and $\beta(S_k)$ are called an alpha value and a beta value, respectively.

[18] FIG. 1 is a timing diagram of the related art MAP decoding. The X-axis represents the flow of time, and especially represents which symbol each processor processes as the time flows. The number of symbols of a forward processor is increased, and the number of symbols of a backward processor is reduced. The slant-lined shading sections represent that the backward processor is in learning. The curved arrows represent the correlation between alpha 5,7 and beta 6,8 values required for the bit decision.

[19] Referring to FIG. 1, two backward processors are used. One backward processor (e.g., the first backward processor 2) performs the learning while the other backward processor (e.g., the second backward processor 3) calculates the beta values required for the bit determination of the decoder.

[20] Specifically, when one MAP decoding starts, the first backward processor 2 performs the learning process from 2L to 1L. During this learning process, the second backward processor 3 is in a standstill state. Here, L represents the length of the sliding window.

[21] Thereafter, the first backward processor 2 calculates the beta values 6 from 1L to 0 and determines bits of the decoder from 1L to 0 using the alpha values 5 from 0 to 1L, which

were previously calculated and stored. During the bit determination of the decoder, the second backward processor 3 performs the learning using symbols from 3L to 2L.

[22] In the next sliding window section, the second backward processor 3 determines the bits of the decoder from 2L to 1L by calculating the beta values 8 from 2L to 1L. During this bit determination of the decoder, the first backward processor 2 performs the learning using symbols from 4L to 3L.

[23] As can be seen in the decoder output block, the bit determination is performed in order from 1L to 0, and from 2L to 1L. A proper order is obtained through a last-in first-out (LIFO) process that stores L outputs at a time and reads them from the end.

[24] The above-described process is explained in detail in 'A.J. Viterbi, "An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes," IEEE Journal on Selected Areas in Communications, vol.16, no.2, Feb. 1998.'

[25] According to the conventional method as described above, the backward processing is performed twice with respect to almost all symbols. As a result, two times backward processing is required for each MAP decoding and this causes the amount of calculation and the power consumption to be increased. Thus, the use time of the radio mobile equipment that operates using a battery is reduced.

[26] Also, in case that only one backward processor is used for reducing the amount of calculation, the decoding time is increased twice as much.

[27] Also, if the learning process, forward processing, and backward processing are performed for the length of L, the characteristic that the coding is completed with ST=0 at a

trellis termination of the turbo codes cannot be sufficiently used. This causes the coding gain of the turbo codes to deteriorate.

[28] Also, though the size of the memory for storing resultant values is small to the extent of depth 60 * width 56 (in case of 3GPP WCDMA turbo encoder), the depth of a generally used memory is much larger than this and this causes the waste of memory.

[29] The above references are incorporated by reference herein where appropriate for appropriate teachings of additional or alternative details, features and/or technical background.

SUMMARY OF THE INVENTION

[30] An object of the invention is to solve at least the above problems and/or disadvantages and to provide at least the advantages described hereinafter.

[31] Another object of the present invention is to provide a device and method of decoding turbo codes that requires a small amount of calculation.

[32] A further object of the present invention is to provide a device and method for decoding turbo codes that uses a small-sized memory.

[33] A further object of the present invention is to provide a device and method for decoding turbo codes that uses one backward processor.

[34] A further of the present invention is to provide a device and method of decoding turbo codes that is suitable for increasing a coding gain.

[35] A further of the present invention is to provide a device and method for decoding turbo codes that reduces the power consumption.

[36] A further of the present invention is to provide a device and method for decoding turbo codes trellis termination at maximum, using one backward processor.

[37] A further of the present invention is to provide a device and method of decoding turbo codes that is suitable for reducing a decoding time.

[38] To achieve these objects and other advantages and in accordance with the purpose of the invention, as embodied and broadly described herein, a MAP decoder in a receiving end that performs an iterative decoding includes a backward processor for calculating first resultant values that are state probability values, after a reference number of a received sequence for an L bit length of the received sequence, and calculating and storing second resultant values that are state probability values, after the reference number of the received sequence for a W bit length of a next received sequence; a forward processor for calculating third resultant values that are state probability values, before the reference number of the received sequence, simultaneously with calculation of the first resultant values; a memory for storing in order the second resultant values and outputting in a reverse order the second resultant values after calculation of the second resultant values; and an output determination module for determining output values of the received sequence using the calculated third resultant values and the outputted second resultant values.

[39] Preferably, the memory writes the second resultant values by alternately using increasing addresses and decreasing addresses and outputs the second resultant values by

alternately using the decreasing addresses in symmetry with the write and the increasing addresses.

[40] Exceptionally, the first resultant values firstly calculated from the received sequence are calculated in the reverse order from bits (i.e., L + the remainder obtained by dividing the length of the received sequence by W) and the second resultant values firstly calculated from the received sequence are calculated in the reverse order from the received sequence for a length of the remainder obtained by dividing a length of the received sequence by W to be stored in the reverse order. Also, the third resultant values firstly calculated from the received sequence are calculated in order from the received sequence for a length of the remainder obtained by dividing the length of the received sequence by W.

[41] Preferably, the output values of the received sequence are determined in a manner that the second resultant values outputted in the reverse order becomes in the same order as the third resultant values calculated in order.

[42] In another aspect of the present invention, a method of performing a MAP turbo decoding in a receiving end that performs an iterative decoding, includes the steps of calculating first resultant values that are state probability values after a reference number of a received sequence for an L bit length of the received sequence and calculating and storing second resultant values that are state probability values after the reference number of the received sequence for a W bit length of a next received sequence; calculating third resultant values that are state probability values before the reference number of the received sequence simultaneously with calculation of the first resultant values; storing in order the second resultant values and

outputting in a reverse order the second resultant values after calculation of the second resultant values; and determining output values of the received sequence using the calculated third resultant values and the outputted second resultant values.

[43] Preferably, the second resultant values are written by alternately using increasing addresses and decreasing addresses and the second resultant values are outputted by alternately using the decreasing addresses in symmetry with the write and the increasing addresses.

[44] Exceptionally, the first resultant values firstly calculated from the received sequence are calculated in the reverse order from bits (i.e., L + the remainder obtained by dividing a length of the received sequence by W) and the second resultant values firstly calculated from the received sequence are calculated in the reverse order from the received sequence for a length of the remainder obtained by dividing a length of the received sequence by W to be stored in the reverse order. Also, the third resultant values firstly calculated from the received sequence are calculated in order from the received sequence for a length of the remainder obtained by dividing the length of the received sequence by W.

[45] Accordingly, the present invention considers the characteristic of trellis termination at the termination of the MAP decoder and, thus, has an effect of obtaining the coding gain.

[46] Preferably, the output values of the received sequence are determined in a manner that the second resultant values outputted in the reverse order become in the same order as the third resultant values calculated in order.

[47] In still another aspect of the present invention, a method of decoding a received sequence using a Maximum A Posteriori (MAP) algorithm, includes the steps of performing a learning by a backward processing for a predetermined length, calculating and storing first resultant values by the backward processing, calculating second resultant values by a forward processing simultaneously with the learning time, and determining a decoding bit output using the second resultant values and the first resultant values stored before the first resultant values.

[48] Preferably, if a length of the backward or forward processing is W, a length of learning is L, a remainder obtained by dividing the length of the received sequence by W is W_0, and N is an integer not less than 1, the learning is performed by the backward processing with symbols of the number of received sequence W_0+NW+L to W_0+NW, the first resultant values by the backward processing with the symbols of W_0+NW to W_0+(N-1)W are stored, the second resultant values by the forward processing with the symbols from W_0+(N-1)W+L to W_0+NW are calculated simultaneously with the learning time, and a decoding bit determination is performed with the second resultant values and the first resultant values calculated and stored from W_0+(N-1)W to W_0+NW.

[49] However, in case that N is 0, the learning is performed by the backward processing with the symbols of the number of received sequence W_0+L to W_0, the first resultant values by the backward processing with the symbols of W_0 to 0 are stored and then the second resultant values by the forward processing with the symbols from 0 to W_0 are calculated simultaneously with the learning start of a next window to calculate the second resultant values. Here, the first resultant values are written through one port of a dual-port RAM (DPRAM) and

read out through another port thereof. Addresses stored or read out through the ports of the DPRAM are increased or decreased for each length of W_0 or W and the addresses stored or read out through the ports of the DPRAM are increased or decreased for each length of W_0 or W in a mutually exclusive manner.

[50] Preferably, the decoding bit output is determined in order.

[51] The objects of the present invention may be achieved in whole or in part by a MAP decoder, including a backward processor that calculates first resultant values of an L-bit length sequence and second resultant values of a W-bit length sequence; a forward processor that calculates third resultant values; a memory that stores the second resultant values in a first order and outputs the second resultant values in a second order that is the reverse of the first order; and an output determination module that determines output values of a received sequence using the third resultant values and the outputted second resultant values. The L-bit length sequence and W-bit length sequence are portions of the received sequence. The first, second, and third resultant values are state probability values. The calculations of the first and third resultant values overlap in time and the calculation of the first resultant values is performed after the calculation of the second resultant values is completed.

[52] The objects of the present invention may be further achieved in whole or in part by a method of performing a MAP turbo decoding. The method includes calculating first resultant values of an L-bit length sequence, calculating second resultant values of a W-bit length sequence, calculating third resultant values, storing the second resultant values in a first order and outputting the stored second resultant values in a second order that is the reverse of the first

order, and outputting decoded values of a received sequence using the third resultant values and the outputted second resultant values. The L-bit length sequence and W-bit length sequence are portions of the received sequence. The first, second, and third resultant values are state probability values. The calculations of the first and third resultant values overlap in time and the calculation of the first resultant values is performed after the calculation of the second resultant values is completed.

[53] The objects of the present invention may be further achieved in whole or in part by a method of turbo-decoding a received sequence using a Maximum A Posteriori (MAP) algorithm. The method includes performing a learning by a backward processing for a predetermined length, calculating and storing first resultant values obtained by the backward processing, calculating second resultant values by a forward processing that overlaps in time with the learning, and determining a decoding bit output using the second resultant values and the stored first resultant values.

[54] Additional advantages, objects, and features of the invention will be set forth in part in the description which follows and in part will become apparent to those having ordinary skill in the art upon examination of the following or may be learned from practice of the invention. The objects and advantages of the invention may be realized and attained as particularly pointed out in the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[55] The invention will be described in detail with reference to the following drawings in which like reference numerals refer to like elements wherein:

[56] FIG. 1 illustrates a timing diagram of the related art MAP decoding; and

[57] FIG. 2 illustrates a timing diagram of the MAP decoding according to the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[58] As described above, the MAP algorithm is an algorithm that estimates the *a posteriori* probability of the information bits $P(d_k=1 \mid y)$ and $P(d_k=0 \mid y)$ and then finally obtains the output of the decoder in the form of a log likelihood ratio (LLR) from the estimated probability of the information. The output of the decoder in the LLR form is described by equation 1.

[59] To obtain the *a posteriori* probability of the information bits, the *a posteriori* probability of the state shift should be obtained with respect to $P(d_k=1 \mid y)$ and $P(d_k=0 \mid y)$, respectively. The respective *a posteriori* probability of the state shift can be obtained by the three multiplication terms. The three multiplication terms with respect to the *a posteriori* probability of the state shift are expressed by equation 2.

[60] Specifically, referring to equation 2, the first term ($\alpha(S_k\text{-}1)$) is a joint probability function of the state S_k-1 that is m' in the received sequence having time indexes from 0 to k-1 and is expressed by the following equation 3.

[61] [Equation 3]

$$\alpha(S_k\text{-}1) = P(S_k\text{-}1 = m', y_{j<k})$$

[62] In equation 3, $\alpha(S_k)$ is a joint probability density function of the state shift where the state S_{k-1} is m' and the state S_k is m in the received sequence $y_{j<k+1}$, having the sequence numbers from 0 to k, and is expressed by the following equation 4.

[63] [Equation 4]

$$\alpha(S_k) = \sum_{m'=0}^{M-1} P(S_{k-1} = m', S_k = m, y_{j<k+1})$$

[64] The second term $\gamma(S_{k-1},S_k)$ is a branch metric relating when a state S_{k-1} is shifted to a state S_k and is a conditional probability function where the next state S_k is m, on condition that the state S_{k-1} is m' and the sequence received at that time is y_k. It is expressed by the following equation 5.

[65] [Equation 5]

$$\gamma(S_{k-1},S_k) = P(S_k = m, y_k \mid S_{k-1} = m')$$

[66] The third term $\beta(S_k)$ is a conditional probability function where the number of the received sequence $y_{j>k}$ is not less than k+1, on condition that the state S_k is m and is expressed by the following equation 6.

[67] [Equation 6]

$$\beta(S_k) = P(y_{j>k} \mid S_k = m)$$

[68] In equation 6, d_k represents the information bit sequence before the turbo encoding and S_k represents the state ($m=\{0,1,..,M-1\}$) of the encoder at the number k of the received sequence. Both d_k and S_k have M kinds of states and the input bit d_k changes the state of the encoder from S_{k-1} to S_k when the number of the received sequences is shifted from k-1 to k.

[69] Specifically, in the MAP decoding, $\alpha(S_k)$ can be obtained by a forward recursion method as in the equation 4 and the calculated $\alpha(S_k)$ is directly used for determining the output bits of the decoder. $\alpha(S_k)$ is performed by one forward (alpha) processor.

[70] Also, $\beta(S_k)$ can be obtained by a backward recursion method as in the equation 6. For $\beta(S_k)$ to be used in obtaining the MAP probability, a predetermined period that is called a learning period is required. After this learning period, $\beta(S_k)$, calculated by one backward (beta) processor, is used for determining the output bits of the decoder.

[71] Hereinafter, $\alpha(S_k)$ and $\beta(S_k)$ are called the alpha value and the beta value, respectively.

[72] FIG. 2 is a timing diagram of the MAP decoding according to the present invention. The backward processor starts the learning with certain symbols from W_0+L to W_0. The learning is performed with a length of L, throughout the MAP decoding. In performing the learning, symbols correspond to the L bits of the received sequence in the same manner as in the related art method. Here, W_0 corresponds to the remainder obtained by dividing the length of the received sequence by W.

[73] Next, the beta values 10 are calculated with the symbols from W_0 to 0, and then stored in the memory. Here, a dual port random access memory (DPRAM) having different input/output ports is used as the memory.

[74] Next, the learning is started with the symbols from W_0+W+L to W_0+W. Then, the beta values 11 are calculated with the symbols from W_0+W to W_0 to be stored in the memory.

[75] Simultaneously with the start of learning with the symbols from W_0+W+L to W_0+W, the forward processor calculates the alpha values 13 with the symbols from 0 to W_0 and determines the output bits of the decoder using the calculated alpha values 13 and the beta values 10 from W_0 to 0, which are stored by the backward processor. The calculation of the alpha 13-15 and beta 10-12 values is performed for the length of W.

[76] Next, the learning process is started with symbols from W_0+2W+L to W_0+2W. Then, the beta values 12 are calculated with the symbols from W_0+2W to W_0+W to be stored in the memory.

[77] Simultaneously with the start of learning with the symbols from W_0+2W+L to W_0+2W, the forward processor calculates the alpha values 14 with the symbols from W_0 to W_0+W and determines the output bits of the decoder using the calculated alpha values 14 and the beta values 11 from W_0+W to W_0, which are stored by the backward processor.

[78] As described above, by repeating the learning, backward processing, forward processing, and decoding, the MAP decoding for one code block (for convenience' sake, it has been called a received sequence, the length of which is called one code block size) is completed.

[79] Meanwhile, in two lower lines of FIG. 2, addresses of the dual port RAM (DPRAM) for storing the beta values that are results of the backward processing are illustrated. The beta values are stored through port A and are read out through port B.

[80] However, the beta values calculated during the learning period are not stored.

[81] The beta values 10 calculated from W_0 to 0 are stored in the order of the addresses decreasing from W_0 to 0, through port A, and then read out in the order of the addresses increasing from 0 to W_0, through port B, so that the read beta values 10 are used for determining the output bits of the decoder along with the alpha values 13 calculated from 0 to W_0.

[82] Also, the beta values 11 calculated from W_0+W to W_0 are stored in the order of the addresses increasing from 0 to W, through port A, and then read out in the order of the addresses decreasing from W to 0, through port B, so that the read beta values 11 are used for determining the output bits of the decoder along with the alpha values 14 calculated from W_0 to W_0+W.

[83] Specifically, ports A and B store and read the beta values calculated for the predetermined length in the order of the increasing or decreasing addresses. In case that the beta values have been stored in the order of the increasing addresses through port A, they are read out in the order of the decreasing addresses through port B. In case that they have been stored in the order of the decreasing addresses, they are read out in the order of the increasing addresses through port B.

[84] That is, ports A and B input/output the values in the order of the increasing or decreasing addresses in a mutually exclusive manner.

[85] If the addresses being used proceed always in the same direction (i.e., if it is repeated that the addresses of port A are decreased from W or W_0 to 0 and the addresses of port B are increased from 0 to W or W_0), the stored beta values are updated with new values before the stored beta values have been read out for use in determining the decoder output. In order to prevent this, the size of the memory for storing the beta values should be increased twice or the method proposed according to the present invention should be used.

[86] According to the present invention, one backward processing starts from W_0+NW+L (N=0,1,2,..) and this start point is increased by W for each subsequent sliding window. Also, the end point of the backward processing is W_0+NW (N=0,1,2,..), and is also increased by W. In performing the learning, symbols correspond to the L bits of the received sequence in the same manner as in the related art method.

[87] $W_0$is determined by the equation N mod W, where N is the length of the received sequence and "mod" (refers to modulo calculation). However, if the result of the modulo calculation is 0, W is used instead.

[88] For example, if the length of the received sequence is 3840 bits and W is 256, W_0 is determined to be 256, which is equal to W. If the length N of the received sequence is 3841 bits, W_0 becomes 1. As W_0 is determined as described above, the final unit of the backward processing will be always W. By doing this, the property of the turbo codes that use the trellis

termination (3GPP TS25.212 V2.2.1 Turbo Coding section, Oct.1999) such as 3GPP WCDMA can be effected at maximum.

[89] This is the same concept that when the convolutional codes using tail bits are decoded by a Viterbi decoder, just the very last trace-back depth of the code block starts from the state 0 and is decoded at a time.

[90] For example, if it is determined that W is 256, one bit remains in the code block of 3841 bits, and it may be considered that the one bit is processed in the very first window according to the present invention or in the very last window. Since the termination characteristic can be used at the end of each code block, a better decoding performance can be obtained by decoding 256 bits, rather than by decoding one bit, using the termination characteristic.

[91] With regard to the size of the memory required for the MAP decoding, the conventional method requires about 60 depth, while the decoder according to the present invention uses 256 depth. However, there may be no difference in implementing the actual circuit since the minimum depth of an internal block RAM of a "Xilinx Virtex" chip used for implementing the present invention is 256.

[92] As described above, the present invention uses only one backward processor and thus the size of circuit and the amount of calculation (i.e., power consumption) can be reduced.

[93] Also, a better decoding performance can be achieved by effecting the characteristic of trellis termination by filling the last window.

[94] Also, since the decoding results can be obtained in order, the memory and the circuit required for the LIFO can be removed and the power consumption can be improved.

[95] Also, by adjusting the read/write addresses of the dual port RAM for storing the beta values, the size of the memory can be reduced by half.

[96] The foregoing embodiments and advantages are merely exemplary and are not to be construed as limiting the present invention. The present teaching can be readily applied to other types of apparatuses. The description of the present invention is intended to be illustrative, and not to limit the scope of the claims. Many alternatives, modifications, and variations will be apparent to those skilled in the art. In the claims, means-plus-function clauses are intended to cover the structures described herein as performing the recited function and not only structural equivalents but also equivalent structures.